

In the Claims

The following listing of the claims replaces all previous listings.

1. (Currently Amended) A software engine for application loading a software application onto a user's machine, wherein a core service of the application is loaded onto the user's machine to enable the application to commence to operate on the user's machine, the engine subsequently loading non-core services of the application according to a priority order determined by the engine, wherein a non-core service is responsible for providing a functionality of the application and corresponds to a user interaction with the application during run-time, and wherein, in response to the user interaction during run-time, corresponding non-core services are designated a top priority in the priority order such that functionality of the application is enabled.
2. (Original) A software engine as claimed in claim 1, wherein the engine is part of the core service and is loaded with the core service.
3. (Original) A software engine as claimed in claim 2, wherein the engine commences operation upon completion of loading of the core service.
- 4.-6. (Canceled)
7. (Currently Amended) ~~An engine~~ A software engine as claimed in claim ~~[[5]]~~ 1, wherein before loading the non-core services they are registered with the engine.
8. (Currently Amended) A software engine as claimed in claim 7, wherein the engine checks ~~[[the]]~~ a registration list of non-core services before loading a requested non-core service.
9. (Original) A software engine as claimed in claim 1, wherein there is provided a cache into which at least one object for the application can be stored.

10. (Original) A software engine as claimed in claim 9, wherein the engine includes a memory management module that keeps track of usage of cached objects; the memory management module being able to de-allocate one or more of the objects.
11. (Original) A software engine as claimed in claim 10, wherein the cache is operative only when the application is on the user's machine.
12. (Original) A software engine as claimed in claim 9, wherein the cache includes an object repository into which the at least one object is placed, and an object description.
13. (Original) A software engine as claimed in claim 12, wherein the object description includes one or more selected from the group consisting of: object reference, object key, reference counter and time stamp.
14. (Original) A software engine as claimed in claim 10, wherein the de-allocation of one or more of the objects includes an arbitrary time offset.
15. (Original) A software engine as claimed in claim 14, wherein if the object description of an object in the object repository has a reference counter equal to zero for a time equal to at least the time offset, the corresponding object description is removed from the object repository.
16. (Original) A software engine as claimed in any one of claim 1, wherein the loading is downloading over the Internet.
17. (Currently Amended) A method of loading a software application onto a user's machine using a software engine, the method including loading onto the user's machine core services of the application to enable the user to interact with the application, and loading non-core services of the application according to a priority order determined by the engine, wherein a non-core service is responsible for providing a functionality of the application and corresponds to a user interaction with the application during run-time, and wherein, in response to the user interaction

during run-time, corresponding non-core services are designated a top priority in the priority order such that functionality of the application is enabled.

18. (Original) A method as claimed in claim 17, wherein the engine is part of the core service and is loaded with the core service.

19. (Original) A method as claimed in claim 18, wherein the engine commences operation upon completion of loading of the core service.

20. (Original) A method as claimed in claim 17, wherein before loading the non-core services they are registered with the engine.

21. (Canceled)

22. (Currently Amended) A method as claimed in claim ~~[[21]]~~ 17, wherein upon interaction with the application by the user, the application requests the engine to load at least one of the non-core services, and the engine checks ~~[[the]]~~ a registration and gives the at least one non-core service top priority for loading.

23. (Canceled)

24. (Canceled)

25. (Original) A method as claimed in claim 17, wherein objects of the application are storable in a cache for reuse.

26. (Original) A method as claimed in claim 25, wherein cached objects are tracked using a memory management module of the engine, which can de-allocate one or more of the objects.

27. (Original) A method as claimed in claim 26, wherein the cache is operative only when the application is on the user's machine.

28. (Original) A method as claimed in claim 25, wherein the objects are placed into an object repository in the cache, together with an object description.
29. (Original) A method as claimed in claim 28, wherein the object description includes one or more selected from the group consisting of: object reference, object key, reference counter and time stamp.
30. (Original) A method as claimed in claim 26, wherein de-allocation includes an arbitrary time offset.
31. (Original) A method as claimed in claim 30, wherein if the object description of an object repository has a reference counter to equal to zero for a time equal to at least the time offset, the corresponding object description is removed from the object repository.
32. (Original) A method as claimed in claim 17, wherein the loading in downloading over the Internet.
33. (Currently Amended) A software engine as claimed in claim 1, further comprising a computer memory management system for use with a software application, the memory management system including a cache, and wherein objects of the application are storable in the cache for reuse.
34. (Currently Amended) A ~~system~~ software engine as claimed in claim 33, wherein the cache is operative only when the application is on the user's machine.
35. (Currently Amended) A ~~system~~ software engine as claimed in claim 33, wherein the objects are placed into an object repository in the cache, together with an object description.

36. (Currently Amended) A ~~system~~ software engine as claimed in claim 35, wherein the object description includes one or more selected from the group consisting of: object reference, object key, reference counter and time stamp.

37. (Currently Amended) A ~~system~~ software engine as claimed in claim 33^[(2)], wherein cached objects are tracked using a memory management module, which can de-allocate one or more of the objects.

38. (Currently Amended) A ~~system~~ software engine as claimed in claim 37, wherein de-allocation includes an arbitrary time offset.

39. (Currently Amended) A ~~system~~ software engine as claimed in claim 38, wherein if the object description of an object repository has a reference counter to equal to zero for a time equal to at least the time offset, the corresponding object description is removed from the object repository.